



Syddansk Universitet

Is Water-Scrum-Fall Reality?

On the Use of Agile and Traditional Development Practices

Theocharis, Georgios; Kuhrmann, Marco; Münch, Jürgen; Diebold, Philipp

Published in:

Product-Focused Software Process Improvement

DOI:

[10.1007/978-3-319-26844-6_11](https://doi.org/10.1007/978-3-319-26844-6_11)

Publication date:

2015

Document version

Early version, also known as pre-print

Citation for pulished version (APA):

Theocharis, G., Kuhrmann, M., Münch, J., & Diebold, P. (2015). Is Water-Scrum-Fall Reality? On the Use of Agile and Traditional Development Practices. In P. Abrahamsson, L. Corral, M. Oivo, & B. Russo (Eds.), Product-Focused Software Process Improvement: 16th International Conference, PROFES 2015, Bolzano, Italy, December 2-4, 2015, Proceedings (pp. 149-166). Springer. Lecture Notes in Computer Science, Vol.. 9459, DOI: 10.1007/978-3-319-26844-6_11

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Is *Water-Scrum-Fall* Reality? On the Use of Agile and Traditional Development Practices

Georgios Theoharis¹, Marco Kuhrmann¹, Jürgen Münch², Philipp Diebold³

¹ University of Southern Denmark, The Mærsk Mc-Kinney Møller Institute & Center for Energy Informatics, Odense, Denmark

g.theoharis84@gmail.com, kuhrmann@mmmi.sdu.dk

² University of Helsinki, Department of Computer Science, Helsinki, Finland

Juergen.Muench@cs.helsinki.fi

³ Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany

philipp.diebold@iese.fraunhofer.de

Abstract. For years, agile methods are considered the most promising route toward successful software development, and a considerable number of published studies the (successful) use of agile methods and reports on the benefits companies have from adopting agile methods. Yet, since the world is not black or white, the question for what happened to the traditional models arises. Are traditional models replaced by agile methods? How is the transformation toward Agile managed, and, moreover, where did it start? With this paper we close a gap in literature by studying the general process use over time to investigate how traditional and agile methods are used. Is there coexistence or do agile methods accelerate the traditional processes' extinction? The findings of our literature study comprise two major results: First, studies and reliable numbers on the general process model use are rare, i.e., we lack quantitative data on the actual process use and, thus, we often lack the ability to ground process-related research in practically relevant issues. Second, despite the assumed dominance of agile methods, our results clearly show that companies enact context-specific hybrid solutions in which traditional and agile development approaches are used in combination.

Keywords: Development Practices, Agile Methods, Software Process, Systematic Literature Review, Comparative Study, Scrum

1 Introduction

Software managers and developers are still on the quest for adequate methods to organize, plan, and direct software projects. Since the early 1970's, software processes are defined to capture and organize knowledge about software and systems development and, since then, a continuously growing number of software development approaches competes for the users' favor. The software process comes in different shapes—be it as rigid plan-driven approach or as slim agile method, and since the 2000's, we find two basic streams. Traditional (or rich) processes aim to address the whole software project lifecycle, e.g., by providing comprehensive guidelines, standardized procedures, project planning templates, and interfaces to further organization processes, while agile methods aim at reducing the software process to its minimum to avoid “bureaucracy”, and to

provide users with only this amount of rules and guidelines that is required to perform a project. Over the past decades, we observed the “rise of agile methods” [8]. Apparently, agile methods provide a better way to address practitioners’ needs, to accelerate software development, and to improve quality and customer satisfaction. Several studies provide evidence regarding the benefits of agile methods, e.g., [6, 9]. Yet, practice shows the world not being fully “agilized”, as there exist settings in which agile methods are either not (fully) applicable or cannot show their strengths. In such situations, the software process is usually adopted to the current context, which, for instance, is shown in our previously conducted study [21] in which we could identify different combination patterns showing a trend toward West’s “Water-Scrum-Fall” [40]. In particular, we found combinations of traditional and agile software development approaches, which was also found in another independently conducted study [39]. Furthermore, surveys on the use of agile methods, such as [17, 38], suggest that even the agile methods appear to be rarely used in their “pure” form. For instance, Diebold et al. [7] conducted a study in which they show how Scrum is used in practice (with a particular focus on variations and respective rationale).

Problem Statement Available studies draw a picture of a diverse process ecosystem in which different development approaches are combined with each other. Therefore, knowledge of the project context, customization settings, and variability operators is required to efficiently assemble a company- or project-specific process. However, different studies reveal considerable gaps in the body of knowledge [4, 24, 25]. As mentioned by Dingsøyr et al. [8], current literature focuses on researching agile methods, yet ignoring the other processes to a large extent. To identify useful strategies to systematically and efficiently combine development approaches for a particular context, understanding of the state-of-the-practice is required. In particular, we miss empirical data on the process use in general, about common combinations, and trends over time.

Objective To investigate the use of different software process models in practice, we analyze published studies that provide data on software processes, their use, potential/implemented combinations, and trends by analyzing how the use of software processes evolved over time. Our research aims to investigate practically applied processes and combinations to lay the foundation for developing appropriate process customization and configuration instruments.

Contribution This paper presents our findings from a literature study on the use of process process models in practice. In a set of 22 selected papers, only five provided quantitative data. Furthermore, data from the annually conducted VersionOne survey [38] were used for comparison. Our findings show software processes mainly used as hybrid approaches that either combine traditional and agile methods or compose project processes from different practices. However, our findings also show absence of detailed knowledge about which approaches are used in particular, how those are combined, and how they relate to the respective company context. Studies that investigate the whole process ecosystem are scarcely to find. Therefore, our study also concludes with the call for more research to generate more and better data on process use to pave the way for developing efficient instruments for process selection, tailoring, and combination.

Outline The reminder of the paper is organized as follows: Section 2 discusses related work and gaps in the current body of knowledge. In Section 3, we describe our research approach, present the results of our study in Section 4, and provide a discussion on the results in Section 5. We conclude the paper in Section 6.

2 Related Work

Agile software development methods were introduced and used in the late 1990's. They gained attention and companies and developers worldwide increasingly adopted them. In February 2001, the *Agile Manifesto*, was published and described the principles of Agile Methods. Since then, agile methods gained increasing attention in practice as well as in research and, consequently, a number of agile methods was proposed. In 2014, Tripp and Armstrong [37] investigated the “most popular” agile methods and found, inter alia, Extreme Programming (XP), Scrum, Dynamic Systems Development Method (DSDM), Crystal, Feature Driven Development (FDD), and Lean development the most frequently used. Beyond such one-time studies, few studies investigate the development over time. For example, a comprehensive perspective on the use of agile methods is provided by Dingsøyr et al. [8], who summarized “a decade of agile software development”, provided an overview of the publication body, and motivate further research toward a sound theoretical framework of agile software development. Dybå and Dingsøyr [9] conducted a literature review on empirical studies on agile software development. They summarized the major approaches, but found most of the research done in the context of XP (with immature teams) and concluded the strength of evidence (still) low. Furthermore, they vote for more rigorous research in the field of agile methods (especially on methods of particular relevance for industry). Such an industry perspective is given by the VersionOne survey [38] that investigates the use of agile methods over time and draws a slightly different picture⁴. Aggregating the VersionOne data, Scrum became the most popular agile method. However, this study also points to a certain trend toward using hybrid and customized approaches, which is also supported by Diebold et al. [7], who study how Scrum is actually implemented in practice.

Going beyond aforementioned agile-focused studies and reviewing the available scientific and “grey” literature on software processes in general, one may conclude that agile methods took over from traditional processes. In fact, literature reflects this trend. For instance, in their 1996-study, Khurana et al. [15] list more than 25 fine-grained “techniques” for software engineering (mapped to different software engineering activities), e.g., prototyping, Fagan inspections, and SSADM [32]. Studies conducted in the early 2000's, e.g., [3, 11] show several of the mentioned techniques still in use, but also show them slowly disappearing and being replaced by (the new) agile methods.

However, within that diversity of traditional and agile methods, another trend came to light: just in 1997, Fitzgerald [10] found 6 out of 8 companies in Ireland using custom or non-formalized development processes—a trend that still can be observed over time (e.g., Reifer [31]) and also in recent studies, e.g., [18]. Among other things, those studies also show that the traditional process models are still of certain relevance.

⁴ A similar study is available by the “Status Quo Agile 2014” study [17]. However, this study does not provide as comprehensive historical data as the VersionOne survey series.

For example, Solinski and Petersen [36] found Scrum and XP the most popular and adopted methodologies. They also found combinations of the classic Waterfall/XP, and Scrum/XP the most common combinations. Recently published studies, e.g., [18, 39] also indicate to a situation in which traditional and agile approaches coexist and make the majority of practically used (mixed/hybrid) approaches.

Therefore, current literature on software processes and their application (patterns) in practice leaves researchers and practitioners alone with a diffuse picture: agile methods gained popularity and the related (scientific) literature is dramatically increasing (cf. [8]), e.g., understanding of benefits of certain practices and analyzing the impact of agile methods. Traditional process models are seemingly vanished from the researchers' to-do lists and, if at all, appear only in the context of process modeling, in domains having special requirements to the software process, e.g., compliance, regulations, and norms (safety & security, medical devices, etc.), as part of surveys, and as subject to discussion why companies might be reluctant to buy-in agile methods (e.g., [26, 37]). However, information about general software process use, local/global trends, and detailed information about combination patterns is missing in literature. The present paper thus aims to fill a gap in literature by providing first steps toward a more comprehensive picture of process use to lay the foundation for further research.

3 Research Design

In this curiosity-driven and exploratory study, we opted for a research design in which we applied instruments from *Systematic Literature Review* process [16]. The core study was conducted by stepwise collecting information starting from a small set of reference publications to form the basic knowledge (snowballing). Based on the harvested publications, we conducted an automatic search in different literature databases to find further publications, which then were used for another snowballing iteration. Finally, the VersionOne survey [38] was chosen for a broader analysis and discussion.

In subsequent section, we detail the research design and explain the different procedures to collect and analyze data.

3.1 Research Questions

To conduct the study, we defined the following research questions:

RQ 1 *Which development approaches are used in practice?* This research question aims at developing a catalog of practically used development approaches (processes, methods, or practices) and to investigate the approaches' use over time.

RQ 2 *Which development approaches are combined in practical use?* In order to answer this research question, we use the initial catalog to study how development approaches are combined in order to address specific software development contexts.

RQ 3 *Are there patterns and trends observable?* This research question aims to investigate patterns of process use to better support the adaptation of agile methods in non-agile environments and trends of process use, i.e., does the region affect the process selection (e.g., due to local standards), or is there any general trend to find.

Table 1: Inclusion and exclusion criteria for the study's SLR parts.

No.	Description
IC ₁	The paper describes a longterm observation of the use of development methods.
IC ₂	The paper surveys practitioners for the use of development methods.
IC ₃	The paper reports on the use of development methods in general, e.g., as secondary study.
IC ₄	The paper is on tools implementing certain methods (infer information about method use).
EC ₁	The paper is a proposal only.
EC ₂	The paper occurred multiple times in the result set.
EC ₃	The paper is not on the use of development methods (out of scope).
EC ₄	The paper is a workshop-, tutorial-, or poster summary.
EC ₅	The paper is not in the domain of software engineering or computer science in general.
EC ₆	The paper reports on teaching experiences using development methods only.
EC ₇	The paper is not available in English (or German).
EC ₈	The paper's full text is not available for download.

3.2 Data Collection Procedures

To get information on the process use in general, we were especially interested into primary studies that we collected using a multi-staged search and selection procedure. The data collection procedure comprised the following steps:

1. Snowballing based on two manually selected reference publications [18, 39].
2. Initial analysis of the found extra four publications: [3, 11, 14, 30].
3. Automated search in five literature databases, complemented by a search using Scopus and Google Scholar as meta-search engines.
4. Snowballing of the publications obtained in the automated search.
5. Using the VersionOne survey [38] to add extra perspectives to the discussion.

The manual paper selection (following the snowballing procedures [16]) was conducted twice. The initial selection was performed on two reference publications [18, 39] and aimed at extending the set of reference publications that built the foundation to construct search queries to support the automated search in different literature databases. The resulting six publications were read with the purpose of finding proper keywords to initiate the automatic search. The second iteration was performed on the publications obtained from the automated search with the purpose of completing the result set.

The automatic search was conducted using the following literature databases: *ACM*, *IEEE*, *Springer*, *Wiley*, and *Elsevier*. To obtain the data, based on the reference publications and the initial snowballing results, we developed different search query candidates that we stepwise tested and refined. We conducted the automated search using the following simple search query: *((("method choice") or "method use") and "software development") and "survey")*⁵

In order to confirm the search results, *Scopus* and *Google Scholar* were used. The outcomes of the automatic search were treated in a rigorous and proven selection process, as described in detail in [19] applying the in-/exclusion criteria from Table 1.

⁵ We conducted several test runs finding this simple string producing the best results. Including/adding keywords like "process" just inflated the result set, yet no extra publication providing quantitative data on process use could be found in these tests.

3.3 Analysis Procedures

For the analysis procedure, we implemented two different approaches. In general all selected publications were included in a spreadsheet in which we classified the respective publications, e.g., for study type, applied instruments, context, and outcomes. We then boiled down all initially selected studies to their essence to prepare the qualitative analysis. For the primary studies (in which we were especially interested) we analyzed the publications for quantitative detailed data on processes, use, combinations, and so forth, and collected these data in a separate spreadsheet for an in-depth analysis.

Due to the low number of studies, we did not perform any statistical tests. Instead, we rely on simple data tables and charts on which we base our discussion and interpretation. Since we found a considerable number of process mentions, we also perform a clustering and a categorization.

3.4 Validity Procedures

To improve the validity of our results, we opted for different strategies. The initial data collection and study selection procedures were performed by two researchers adopting the schematic procedures from previously conducted studies [19, 20]. That is, data collection and selection was initially performed independently using previously defined spreadsheet templates, followed by a series of workshops to perform the final selection and classification. The finally selected papers and initial analysis results were then given to another team of researchers to review, confirm and/or revise the results, and to develop a joint conclusion.

4 Study Results

In this section, we present the results of the study. In Section 4.1, we give an overview of the study population. Sections 4.2 – 4.4 present the results from the literature review and, finally, Section 5 presents a discussion.

4.1 Study Population

In this section, we summarize the study population. The different search procedures resulted into 473 papers from which 22 papers were selected for consideration. Finally, five of the 22 papers provided quantitative data, i.e., survey results on process use, and were selected for an in-depth analysis. These studies were conducted in Germany (2006, 2006, 2013), USA (2014), and Brunei (1998). The remaining papers were considered for the qualitative analysis only. Table 2 provides a summary of the final result set.

4.2 RQ1: Which Development Approaches are used in Practice?

In this section, we provide an overview of the review findings. The five selected papers mention 62 different development approaches in total⁶. Among the named methods,

⁶ Due to space limitations, the study data set including the full list of mentioned processes is available for download here: <http://goo.gl/bUB3Tr>

Table 2: Overview of the papers obtained from the literature search (Instruments: “Q” – questionnaire, “I” – interview, “SLR” – systematic literature review, “M” – miscellaneous instruments).

Paper	Year	Scope in Time		Study Type		Instruments	Quantitative Data
		Snapshot	Trend	Primary	Secondary		
[39]	2014	X		X		Q	✓
[18]	2013	X	(X)	X		Q	✓
[3]	2006	X		X		I, Q	✓
[11]	2006	X		X		Q	✓
[14]	2003	X		X		M	
[30]	2002	X		X		Q	
[35]	2014	X		X		Q	
[36]	2014	X		X		Q	
[27]	2014	X		X		Q	
[5]	2013	X		X		Q	
[26]	2013	X		X		Q	
[37]	2013	X		X		Q	
[22]	2013	X		X	X	M	
[34]	2013	X			X	SLR	
[8]	2012		X		X	SLR	
[23]	2010	X		X	X	Q, SLR	
[28]	2009	X		X		M	
[33]	2008	X		X		Q	
[9]	2008	X			X	SLR	
[12]	2003	X			X	SLR	
[29]	1998	X			X	SLR	✓
[2]	1983	X		X		Q	

Table 3: Mentions of categorized process types (n : number of participants, mentions: number of mentioned processes in total, *: incomplete/not available raw data).

Paper	Year	n	Mentions	Ratio	Traditional	Agile	Technique	Unclassified
[39]	2014	153	316*	2.07	66%	88%	52%	0%
[18]	2013	37	93	2.51	92%	119%	5%	35%
[11]	2006	65	199	3.06	109%	26%	103%	68%
[3]	2006	217*	180	0.83	28%	3%	3%	49%
[29]	1998	36	54	1.50	33%	0%	72%	44%

we find representatives of the traditional development models (e.g., Waterfall, RUP, V-Model), a number of agile methods (e.g., Scrum, Kanban, XP), and also a considerable share of non-standardized, very specific, and/or “legacy” approaches, such as SSADM [32] or Jackson’s System Design [13].

As we found a large number of methods and also varying data presentation, we decided to classify the mentioned approaches as traditional process model, agile method, technique (self-contained method that is neither traditional nor agile, e.g., prototyping), unclassified (if an approach could not be assigned to one of the other classes, or if the paper just states “other”, “custom”, or “don’t know”). Furthermore, due to the

heterogeneous data formats, we normalized the data as relative numbers (percentage of mentions related to the respective n) that is shown in Table 3. This table shows that companies usually use multiple development approaches, which becomes obvious due to the percentage rate larger than 100%. However, the SUCCESS study [3] has to be considered an outlier, as the total mentions are below 100%. As we don't have access to the raw data, we cannot reproduce this phenomenon, yet, since 49% of the mentions are categorized as "unclassified", we cannot conclude to what extent, e.g., agile and traditional approaches are used and/or combined. Especially the *3ProcSurvey* [18] and the IOSE-W² study [11] show that companies use multiple processes. For example, in the *3ProcSurvey*, agile methods account for 119% of the mentions, i.e., even in one project different agile methods are used in combination.

The most frequently mentioned development models across all studies are: Scrum (61), the Waterfall (54), the Rational Unified Process (RUP; 44), and the Agile Unified Process (43). V-shaped processes are present in different variants and account for 105 mentions in total. However, these numbers present the mentions without paying attention to the development over time. For example, in recent studies, RUP accounts for 5.9% [39] and 16.2% [18], whereas for the German context [21] we found a certain trend: in 2006, RUP accounts for 41.5% of all mentions and in 2013, RUP made only 16.2%. Furthermore, and all mentions in 2013 referred to company-specific variants. That is, available studies indicate that—although RUP contributes a high number of mentions—the "classic" (standard/non-customized) RUP is losing its relevance. Further trends are discussed together with observed pattern in Section 4.4.

4.3 RQ2: Which Development Approaches are combined in Practical Use?

In this section, we analyze the combination of the different approaches. The numbers presented in Table 3 show that companies instrument different approaches and use them in combination. However, only one study [18] provides data of sufficient detail to present and discuss the combinations. Figure 1, based on [21], shows the combination of the different development approaches for Germany. It reveals two major clusters and, when analyzing these clusters, shows that the German standard software process V-Modell XT is often combined with Scrum, and Scrum itself is normally used in combination with Kanban and XP. Therefore, based on [21], we can name the following "standard" combination pattern for Germany: *V-Modell XT + (Scrum + (Kanban + XP))*. In other words: companies use the national standard to provide a general management framework (and to comply with certain compliance requirements), and embody the generic process with concrete methods. Furthermore, as even Scrum is fairly generic thus requiring some embodiment [7], companies use Kanban and XP practices for this embodiment.

As mentioned before, we only have sufficient data from the *3ProcSurvey* [18] to allow for such a detailed analysis. However, the numbers from Table 3 suggest a similar picture for other regions, e.g., for USA [39], the "classic" Waterfall, Scrum, and the Agile Unified Process account for a high share of mentions thus suggesting a similar picture in which the V-Modell XT (Germany) is replaced by the Waterfall model as

Participant	Rational Unified Process (Standard)	Rational Unified Process (modified)	Open Unified Process	V-Modell XT (Standard)	V-Modell XT (modified)	V-Modell 97	W-Modell	Hermes	eXtreme Programming	Scrum	Kanban	Crystal	Feature Driven Development (FDD)	None	Other
1				x						x					
2										x	x				
3					x					x					
4															
5				x	x				x	x					x
6															x
7				x	x					x	x				
8				x						x					
9	x			x	x	x			x	x					
10										x	x				
11					x	x				x					
12										x	x				
13				x						x					
14	x				x	x				x	x				
15									x	x					x
16					x					x					
17															x
18										x					
19														x	
20							x							x	
21	x				x				x	x					
22									x	x	x				x
23									x	x					
24						x			x	x					x
25					x		x								
26	x				x								x		
27					x					x					
28									x	x	x	x			
29					x					x					
30									x	x	x				
31	x				x	x				x					
32					x										
33	x				x					x					
34														x	
35															x
36										x					x
37													x		x

V-Modell XT (Standard)					
V-Modell XT (modified)	3				
V-Modell 97	1	4			
Xtreme Programming	2	3	2		
Scrum	5	12	5	9	
Kanban	2	2	1	3	8

Fig. 1: Combination of development approaches (from [21]). The left part shows the use of approaches per respondent revealing two major clusters, and the right part illustrates the combination of the different approaches within these clusters.

overall management framework in which Scrum is embedded (this situation is often referred to as the *Water-Scrum-Fall* [40])⁷.

In a nutshell, although we lack detailed information and raw data about the actual combination of the different approaches, we find indication to a mixed application of traditional and agile approaches. Based on our findings from [18] and available numbers from the other studies, we hypothesize that companies, especially large ones, use a traditional process as reference model to define the basic management procedures and interfaces to the other company- and business processes, while particular project teams utilize a pool of (agile) methods and practices to run development projects. Yet, confirmation of this hypothesis remains subject to future work.

⁷ Although [27] supports this assumption, this study was excluded from the quantitative analysis, as the authors rejected papers not explicitly dealing with agile methods, and summarized all non-agile approaches under “custom or other”. We thus have insufficient information about what processes are eventually meant and how those could affect our study results.

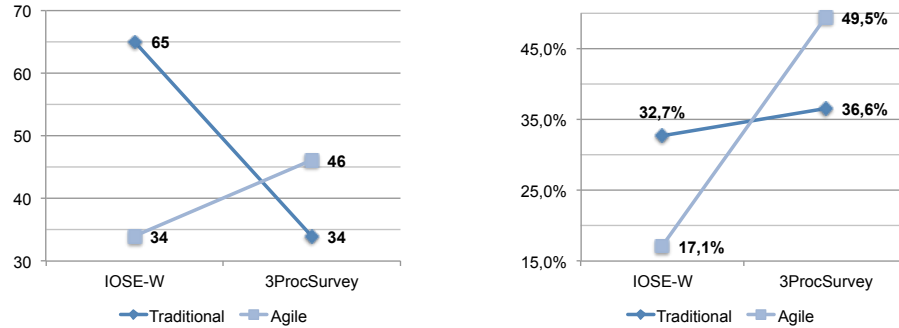


Fig. 2: In-/decrease in use of different process classes (based on the IOSE-W² study from 2006 and the 3ProcSurvey from 2013). The left part shows the absolute numbers and shows the decreased mentions regarding traditional processes and the “rise” of agile methods. However, the right part shows that both kinds of processes are increasingly used (in combination).

4.4 RQ3: Are there Patterns and Trends Observable?

This research question aims at identifying trends in process use. Since software processes are proposed for decades, the question for the relevance occurs. Are approaches proposed in the mid 1990’s still relevant? Or are those approaches dispensable, as they are increasingly replaced by agile methods. Again, due to the absence of longitudinal studies that monitor process use over time, we hardly can provide reliable conclusions. Only for Germany there are three independently conducted studies providing numbers for a period of seven years. In [21], the results from [3, 11, 18] were analyzed for trends in process use. The analysis revealed that although agile methods gained popularity, at the same time, traditional processes were increasingly used as well (Figure 2).

Extending this discussion to the other papers analyzed in the present study, this picture is consistent with the finding that companies usually use mixed-method approaches to organize and operate their projects. For example, [18] showed that the RUP is—if at all—only used in tailored variants. That is, the standard RUP is not used anymore. Comparing this finding to the USA-based study [39], RUP accounts for only 5.9% of all mentions (whereas we don’t have information if this number refers to the standard RUP, tailored variants, or both). Nevertheless, we see “dinosaurs” like the Waterfall model still in use (because of its simplicity regarding general project organization). Furthermore, the combination of different approaches does not only happen between traditional and agile approaches. As, for example, the VersionOne survey [38] shows, even agile methods are used in combination (an in-depth analysis, also consistent with [17], shows that the building blocks used for combination are the individual practices). The VersionOne data shows Scrum champion the other agile methods (56% in 2014), however, the data also shows a stable, yet slightly increasing, number of mixed approaches. Moreover and in line with Diebold et al. [7], we further hypothesize that even the majority of the Scrum users does not apply Scrum by the letter rather than in a customized fashion.

Therefore, we conclude that hybrid approaches that include traditional and agile approaches shape the today's "standard process ecosystem." The "traditional part" is given by a (national) standard, or by a domain- or context-specific standard, e.g., norms like ISO/IEC 26262. The remaining part is given by a context-specific combination of different agile methods or practices. And here, all studies show one trend: Today, Scrum, Kanban, and XP (practices) can be considered the dominating approaches.

5 Study Summary & Discussion

In this section, we summarize and discuss our findings. Furthermore, we review the study with regards to the threats of validity.

Summary & Discussion Although only five papers from the result set provided sufficient information to analyze the general use of software process models over the years (cf. Table 2), the remaining papers provide useful insights as well. Aggregating all data found, we can draw the following picture: in practice, we find hybrid approaches to be the favored solutions. In particular, [7, 38] and [17] showed that even the agile methods are used in combination, and our previously conducted study [18] as well as further independently conducted research [39] provided extra evidence that agile and traditional approaches are used in a mixed-method approach. Among all reviewed studies, we could find a clear trend toward adopting Scrum to the organization's software development [18, 38, 39]. However, we also found some indication that Scrum is often used in combination with further methods [17, 38], and could provide a combination matrix (Figure 1, cf. [21]). Other than expected, Lean Development and DevOps (so far) play no role. However, the current data does not allow for any conclusions thus requiring further research to integrate those new emerging movements into the analysis.

Several industry-related studies showed a certain reluctance of the management to buy-in agile methods [26, 37], which we use as basis to assume that the use of hybrid approaches is, inter alia, caused by this reluctance. For example, Tripp and Armstrong [37] analyzed the motivation to adopt agile methods and found some relevant drivers. Furthermore, they analyzed the perceived impact of 12 selected project management and development-related practices finding those practices negatively correlating with (general) project management, but positively correlating with software development (factor: software quality). Murphy et al. [26] analyzed the use and perception of agile software development at Microsoft. Their major findings include that Scrum is considered the most favorite development approach. However, the study's respondents also considered the suitability of agile methods for large projects critical (substantial communication effort and overhead, reluctance of the management to accept the agile approach). On the other hand, Lagerberg et al. [22], and Solinski and Petersen [36] also found positive impacts of adopting agile practices in large companies, e.g., improved knowledge sharing, increased project visibility, and reduced need for other coordination instruments. Melo et al. [5] conducted a study in Brazil in which they analyzed the evolution of agile software development from the perspectives of education, research, and application in industry and found that the management-related agile practices are either adopted to a large extent or (completely) rejected, while the development-related practices made their way into the projects.

Therefore, we assume that managers prefer the structured planning-oriented way of working, which they need to perform estimation, planning, and controlling tasks, while developers prefer the freedom that agile methods offer the teams. Furthermore, we consider the often missing organizational interface of agile methods (e.g., procurement and contracting procedures, sub-contractor management, and enterprise resource planning) another reason for management's reluctance toward agile methods. Agile methods usually focus on the system under development, management of features or requirements, development and (code-related) quality assurance practices, and the management of change. Organizational interfaces to the company level are, often provided by the traditional approaches.

Hence, integrating traditional and agile approaches into a mixed-method or hybrid approach offers some benefits: managers get their "safe" environment and developers get their freedom and slim development methods. Our study reveals some indication [17, 38] that project teams increasingly compose a "best of" from different practices—quite often on a per-project basis [18]. That is, practices became the building blocks of assembling company- or project-specific software processes. However, information about particular practices used in combination, their linking to the hosting organization, or the exact way of performing those combinations is—if at all—barely available, and reveals a significant gap in the current publication body.

Threats to Validity In the following, we critically discuss our study results regarding these threats. The major threat emerges from the low number of papers selected for the study, and the often low data quality (either due to blurry/reduced data presentation or due to absence of raw data) of the studies analyzed. Therefore, we could present precise data from only one region (Germany), but had to partially infer and normalize data from other studies. Furthermore, since we rely on independently conducted studies, we have no control about the respective data quality. We refer to studies that were conducted using surveys and interviews, i.e., these studies may introduce extra threats to validity by relying on subjective and potentially non-generalizable data. This situation results in an assumption-based line of argumentation that we, however, backed-up by referring to further independently conducted studies, and by performing data analysis and interpretation independently in two teams of researchers. Although we found similar trends, due to the eventually low number of analyzed studies, we therefore have to consider our findings—especially their generalizability—critical.

Another threat to validity is the study selection as such. Due to the blurry terminology that suffers from heterogeneity and massive overloading, after several test runs, we limited the automatic search to "method" and related terms. Yet, the term "development method" has to be considered of insufficient precision to address the entire field. Therefore, even as this term is common in the researched field, limiting the search this way introduces a threat that we addressed by manually conducted search and selection procedures. However, according to Badampudi et al. [1], who found the efficiency of snowballing comparable to automatic searches, we consider our result set suitable for this study. Still, the finally selected result set needs to be considered with care, as we have no knowledge about further publications not triggered by the search and selection procedures applied in this study, and non-scientific studies, e.g., blog posts and forum entries.

6 Conclusion

In this paper, we studied literature for the use of different software development approaches over time. For this, we conducted a literature study in which we applied different instruments from the SLR process to obtain relevant publications. From the different search and selection procedures, we selected 22 publications for inspection and, furthermore, collected quantitative data from five papers from out of this set.

Our findings show a diverse picture in which a variety of different development approaches is used that range from “old-fashioned” traditional approaches to agile methods. Furthermore, our findings show different approaches usually used in combination, i.e., mixed- or hybrid approaches represent the common model of use, and the combination includes “traditional-agile” as well as “agile-agile” approaches. Among the different combinations, Scrum, the classic Waterfall model, and V-shaped processes account for the majority of the mentions. Since we can rely our findings on a limited data basis only, we also extended our investigation to further independently conducted studies from the field of agile methods. In particular, we reviewed the VersionOne survey data and determined trends from the available data. These data confirm our findings that Scrum became the most popular agile method, and that agile methods are adapted and combined with other processes (traditional and agile) [7, 38].

From our findings, we conclude that the *Water-Scrum-Fall* or similar combinations became reality. We discussed possible reasons and argue that the main driver for this kind of integration is the expectations of different stakeholder groups toward the “optimal” software development approach. Project managers require some stability to perform the usual project management tasks, such as estimation, planning, or controlling. Furthermore, project managers also have the responsibility to align projects with the respective company strategy, i.e., projects must interface further (business) processes like human resource management, sales, contracting, and so forth. Since agile methods usually address only system- or development-related tasks, such interfaces are missing often. Hence, traditional approaches are used to provide a basic structure and a framework for project organization and to provide interfaces to the respective company. Then again, developers ask for approaches that—on the one hand—support the development related tasks, but—on the other hand—provide extensive freedom to select the best practices for the respective situation. That is, the hybrid method in which traditional and agile approaches are combined seemingly provides the “win-win” situation.

However, our findings give only indication. Although we could find some trend, still, the limited data does not close the gap that we still miss a clear understanding of how to perform such integration in a structured manner. In particular, the available data does not allow for crafting detailed combination patterns, and the data does only partially allow for investigating the rational behind the approaches’ combination, as for instance in [18] we found project managers making the decision based on experiences and preferences on a per-project base, but couldn’t confirm nor generalize this finding by the other studies, as the required data was not available. Nonetheless, such an understanding is of *growing importance*, as current software development does not only include fast development and release cycles, but increasingly addresses cyber-physical systems that have high requirements, e.g., regarding safety and security. Thus, we must pay special attention to the respective norms and standards that are of similar com-

plexity as traditional software process models. Therefore, discovering approaches to systematically, efficiently, and reliably integrating agile and traditional software development approaches, might also pave the way toward agile software development in critical domains⁸.

Future Work Although based on limited data, our findings have some impact and show some future research directions. The presented study reveals a gap in literature awaiting more work—we ourselves were not able to fully answer all our research questions to full extent. Therefore, our study is a call for action to conduct further research. For this purpose, our study provides initial findings and an initial combination matrix showing how different approaches are used in combination. Furthermore, our findings lay the foundation for further empirical research to gain better insights into the practical application and relevance of certain development approaches. Since we found some confirmation of the *Water-Scrum-Fall* combination pattern, we need to pay more attention to this customization setting that will, eventually, also affect the field of software process improvement (SPI). In [20], we could identify SPI in the context of small-to-medium enterprises and agile software development as “hot topics” (also supported by the more practitioner-oriented surveys on agile methods [17,38]). For this, a better understanding about the different strategies to assemble hybrid approaches for software development promises some benefits, e.g., better addressing and balancing stakeholder requirements, efficient process design, and so forth.

In general the results of this exploratory study show that there is less material published on the combination of software development approaches than expected. Therefore, we would like to motivate the process community to start thinking of how to address this issue, to collect more knowledge, and to build a database that allows for theory development as well as for supporting practitioners in efficiently developing hybrid approaches best fitting their actual requirements. In addition, further data sources and studies have to be collected and analyzed to round out the big picture of process use and to collect more evidence.

Acknowledgements

This research was partially carried out and supported by a Software Campus project (BMBF 01IS12053) funded by the German Ministry of Education and Research.

References

1. D. Badampudi, C. Wohlin, and K. Petersen. Experiences from using snowballing and database searches in systematic literature studies. In *International Conference on Evaluation and Assessment in Software Engineering*, pages 17:1–17:10. ACM, 2015.

⁸ This need is also supported by the just recently published GULP study (<https://goo.gl/RciNpy>) in which authors come to the conclusion that projects will be increasingly operated following a hybrid approach in future (population: 114 IT experts, mainly freelancers and project management consultants; study region: Germany).

2. L. Beck and T. Perkins. A survey of software engineering practice: Tools, methods, and results. *IEEE Transactions on Software Engineering*, SE-9(5):541–561, 1983.
3. R. Buschermöhle, H. Eekhoff, and B. Josko. *SUCCESS and failurE of hard- and Software projectS (SUCCESS)*. BIS-Verlag der Carl von Ossietzky Universität Oldenburg, 2006.
4. D. D. de Carvalho, L. F. Chagas, A. M. Lima, and C. A. L. Reis. Software process lines: A systematic literature review. In *Software Process Improvement and Capability Determination*, volume 477 of *CCIS*, pages 118–130. Springer, 2014.
5. C. de O. Melo, V. Santos, E. Katayama, H. Corbucci, R. Prikladnicki, A. Goldman, and F. Kon. The evolution of agile software development in brazil. *Journal of the Brazilian Computer Society*, 19(4):523–552, 2013.
6. P. Diebold and M. Dahlem. Agile practices in practice: A mapping study. In *International Conference on Evaluation and Assessment in Software Engineering*, EASE '14, pages 30:1–30:10, New York, NY, USA, 2014. ACM.
7. P. Diebold, J.-P. Ostberg, S. Wagner, and U. Zendler. What do practitioners vary in using scrum? In *International Conference XP*, pages 40–51. Springer, 2015.
8. T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe. A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6):1213–1221, 2012. Special Issue: Agile Development.
9. T. Dybå and T. Dingsøyr. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9–10):833 – 859, 2008.
10. B. Fitzgerald. The use of systems development methodologies in practice: a field study. *Information Systems Journal*, 7(3):201–212, 1997.
11. M. Fritzsche and P. Keil. Kategorisierung etablierter vorgehensmodelle und ihre verbreitung in der deutschen software-industrie. Research Report (in German) TUM-I0717, Technische Universität München, 2007.
12. E. Georgiadou. Software process and product improvement: A historical perspective. *Cybernetics and Systems Analysis*, 39(1):125–142, 2003.
13. M. A. Jackson. A system development method. In *Tools and notions for program construction: An advanced course*, pages 1–25. Cambridge University Press, 1982.
14. C. Jones. Variations in software development practices. *IEEE Software*, 20(6):22–27, 2003.
15. M. Khurana, Z. He, I. Court, M. Ross, G. Staples, and D. Wilson. Software quality practices – an empirical study. *Software Quality Journal*, 5(2):75–85, 1996.
16. B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE-2007-01, Keele University, 2007.
17. A. Komus, M. Kuberg, C. Atinc, L. Franner, F. Friedrich, T. Lang, A. Makarova, D. Reimer, and J. Pabst. Status quo agile 2014, 2014.
18. M. Kuhrmann and D. M. Fernández. Systematic software development: A state of the practice report from germany. In *International Conference on Global Software Engineering*. IEEE, 2015.
19. M. Kuhrmann, D. M. Fernández, and M. Tiessler. A mapping study on the feasibility of method engineering. *Journal of Software: Evolution and Process*, 26(12):1053–1073, 2014.
20. M. Kuhrmann, C. Konopka, P. Nellemann, P. Diebold, and J. Münch. Software process improvement: Where is the evidence? In *International Conference on Software and Systems Process*. ACM, 2015.
21. M. Kuhrmann and O. Linssen. Vorgehensmodelle in deutschland: Nutzung von 2006-2013 im überblick. *MAW-Rundbrief*, 39:32–47, 2015.
22. L. Lagerberg, T. Skude, P. Emanuelsson, K. Sandahl, and D. Stahl. The impact of agile principles and practices on large-scale software development projects: A multiple-case study of two projects at ericsson. In *International Symposium on Empirical Software Engineering and Measurement*, pages 348–356. ACM, 2013.

23. G. Lee and W. Xia. Toward agile: An integrated analysis of quantitative and qualitative field data. *MIS Quarterly*, 34(1):87–114, 2010.
24. T. Martínez-Ruiz, F. García, M. Piattini, and J. Münch. Modelling software process variability: an empirical study. *IET Software*, 5(2):172–187, 2011.
25. T. Martínez-Ruiz, J. Münch, and M. Piattini. Requirements and constructors for tailoring software processes: A systematic literature review. *Software Quality Journal*, 20(1):229–260, 2010.
26. B. Murphy, C. Bird, T. Zimmermann, L. Williams, N. Nagappan, and A. Begel. Have agile techniques been the silver bullet for software development at microsoft. In *International Symposium on Empirical Software Engineering and Measurement*. ACM/IEEE, 2013.
27. E. Papatheocharous and A. S. Andreou. Empirical evidence and state of practice of software agile teams. *Journal of Software: Evolution and Process*, 26(9):855–866, 2014.
28. K. Petersen and C. Wohlin. A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of Systems and Software*, 82(9):1479–1490, 2009.
29. M. Rahim, A. H. Seyal, and M. A. Rahman. Use of software systems development methods an empirical study in brunei darussalam. *Information and Software Technology*, 39(14–15):949 – 963, 1998.
30. D. Reifer. How good are agile methods? *IEEE Software*, 19(4):16–18, 2002.
31. D. Reifer. Is the software engineering state of the practice getting closer to the of the art? *Software, IEEE*, 20(6):78–83, 2003.
32. G. B. Rose. SSADM – the open methodology. In *IEE Colloquium on an Introduction to Software Design Methodologies*, number Ref. No: 1991/181, pages 6/1–6/5. IET, Dec 1991.
33. O. Salo and P. Abrahamsson. Agile methods in european embedded software development organisations: a survey on the actual use and usefulness of extreme programming and scrum. *IET Software*, 2(1):58–64, 2008.
34. M. Senapathi and A. Srinivasan. Sustained agile usage: A systematic literature review. In *International Conference on Evaluation and Assessment in Software Engineering*, pages 119–124. ACM, 2013.
35. M. Senapathi and A. Srinivasan. An empirical investigation of the factors affecting agile usage. In *International Conference on Evaluation and Assessment in Software Engineering*, pages 1–10. ACM, 2014.
36. A. Solinski and K. Petersen. Prioritizing agile benefits and limitations in relation to practice usage. *Software Quality Journal*, pages 1–36, 2014.
37. J. Tripp and D. Armstrong. Exploring the relationship between organizational adoption motives and the tailoring of agile methods. In *Hawaii International Conference on System Sciences (HICSS)*, pages 4799–4806, 2014.
38. VersionOne. State of agile survey. Available from: <http://www.versionone.com/agile-resources/more-resources/blogs/>, 2006-2014.
39. L. Vijayasarathy and C. Butler. Choice of software development methodologies - do project, team and organizational characteristics matter? *IEEE Software*, (99):1ff., 2015.
40. D. West. Water-Scrum-Fall is the reality of agile for most organizations today. Technical report, Forrester, 2011.